SE 2AA4 Winter 2007

Software Design Exercise 2

Instructor: William M. Farmer

Assigned:	26 January 2007
Files due:	9 February 2007
Lab report due:	16 February 2007
Revised:	8 February 2007

The purpose of this software design exercise is to write a Java program that creates, uses, and tests a data structure (represented as an object) that stores a single two-dimensional vector. The program will consist of three module specifications (represented as classes).

Background

A *vector* is a mathematical entity that has direction and magnitude. A two-dimensional vector is represented by a pair $V = (a, b)$ of real numbers where a and b are the x- and y-coordinates of V. Given a real number r and two vectors $V_1 = (a_1, b_1)$ and $V_2 = (a_2, b_2)$, the *scalar multiple* of r and V_1 is the vector $rV_1 = (ra_1, rb_1)$ and the *sum* of V_1 and V_2 is the vector $V_1 + V_2 = (a_1 + a_2, b_1 + b_2)$.

Step 1

Write a first module specification that specifies a vector object. It should be a Java class named `Vector` that contains the following public methods:

- A constructor

  ```
  Vector(float a, float b)
  ```

  that constructs an object of type `Vector` that stores a representation of the vector (`a`,`b`).

- Two selectors

  ```
  float getX()
  ```

  and

  ```
  float getY()
  ```

  that return the x and y coordinates of the stored vector, respectively.

- A mutator

  ```
  void mul(float r)
  ```

  that changes the stored vector V to the scalar multiple of `r` and V.

- A mutator

  ```
  void add(float a, float b)
  ```

  that changes the stored vector V to the sum of V and the vector (`a,b`).

Step 2

Write a second module specification that specifies an enhanced vector object that includes the components of a vector object and several additional methods derived from the methods of the vector object. It should be a Java class named `VectorPlus` that is a subclass of `Vector` and contains the following five public methods:

- A constructor

  ```
  VectorPlus(float a, float b)
  ```

  that constructs an object of type `VectorPlus` that stores a representation of the vector (`a,b`). The constructor should be defined using the constructor of the superclass `Vector` with the `super` constructor.

- A selector

  ```
  float getMagnitude()
  ```

  that returns the length of the stored vector.

- A selector

  ```
  float getAngle()
  ```

  that returns the angle from the x-axis to the stored vector measured in radians.

- A predicate

  ```
  boolean isOrthogonal(float a, float b)
  ```

  that returns true iff the stored vector is orthogonal to the vector (`a,b`).

- Two mutators named

  ```
  void setX(float r)
  ```

  and

  ```
  void setY(float r)
  ```

  that change the x-coordinate and the y-coordinate of the stored vector to `r`, respectively.

Step 3

Write a third module specification that tests the first and second module specifications together. It should be a Java class named `TestVectorPlus` that imports the class `VectorPlus` and contains one or more public class methods. Record the results of using this module specification to test the methods in the other two module specifications.

Step 4

Submit the three files `Vector.java`, `VectorPlus.java`, and `TestVectorPlus.java` using subversion. E-mail the `Vector.java` file to your assigned partner (your partner will be posted on the course web site on Thursday, February 8, 2007.). Your partner will likewise e-mail his or her `Vector.java` file to you. You must finish this step no later than 23:59 on Friday, February 9, 2007.

Step 5

After you have received your partner's `Vector.java` file, replace your `Vector.java` file with your partner's. Do not make any modifications to any of the code. Run your test module specification and record the results.

Step 6

Write a report that includes the following:

1. Your name and MAC ID.
2. Your `Vector.java`, `VectorPlus.java`, and `TestVectorPlus.java` files.
3. Your partner's `Vector.java` file.
4. The results of the test of your `Vector.java` and `VectorPlus.java` files.
5. The results of the test of your partner's `Vector.java` file and your `VectorPlus.java` file.

6. A discussion of the test results and what you learned doing the exercise. List any problems you found with (1) your program, (2) your partner's `Vector.java` file, and (3) the specification of the three module specifications. Discuss how these problems could have been avoided.

7. A copy of the part of your log book relevant to this lab exercise.

The lab report is due no later than the beginning of the tutorial session on Friday, February 16.

Notes:

1. Please put your name and MAC ID at the top of each of your source files.

2. Your program must work on birkhoff when compiled by `javac`.

3. If your partner fails to provide you with a copy of his or her `Vector.java` file by the deadline, please tell the instructor via e-mail as soon as possible.